PATENT APPLICATION
Docket No. MS1-1556US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of		)
		)
	Mollicone et al.	)
		)
Serial No.:	10/632,437	) Appeal No.
		)
Confirmation No.	2858	)
		)
Filed:	August 1, 2003	)
		)
For:	Conversion of Structured Documents	)
		)
Examiner:	Botts, Michael K	)

The Honorable Commissioner of Patents
Mail Stop Appeal Brief - Patents
P.O. BOX 1450
Alexandria, VA 22313-1450

BRIEF OF APPELLANT

The Applicant has filed a timely Notice of Appeal from the action of the Examiner in finally rejecting all of the claims that were considered in this application. This Brief is being filed under the provisions of 37 C.F.R. § 1.192. The Filing Fee, as set forth in 37 C.F.R. § 1.17(c), is submitted herewith.

TABLE OF CONTENTS

REAL PARTY IN INTEREST

The real party in interest is Microsoft Corporation, by way of assignment from Mollicone et al, who is the named inventive entity and is captioned in the present brief.

RELATED APPEALS AND INTERFERENCES

None.

STATUS OF CLAIMS

Claims 1-26 are pending and are the subject of this appeal.

STATUS OF AMENDMENTS

An amendment after final was offered to correct typographical errors to claims 22 and 23, but was not entered as requiring a new search, even though the amendment clearly has correct antecedent basis and follows the language of independent claim 20, from which, each of these claims depend.

SUMMARY OF THE CLAIMED SUBJECT MATTER

This invention relates to upgrading documents so that the documents are compatible with a version of a document processing mechanism that is used to process the documents. In a more particular implementation, this invention relates to upgrading arbitrary markup language documents so that the documents match a version of a document processing mechanism used to display and edit the markup language documents. *See Application, Page 1, Lines 2-8.*

According to one exemplary implementation, a method is described for upgrading documents for processing by processing functionality. The method includes: (a) inputting a structured document having particular features associated therewith into a particular version of the processing functionality; (b) determining whether each of the particular features matches a set of expected features associated with the particular version of the processing functionality; and (c) modifying the particular features of the input structured document so that the particular features match the set of expected features to thereby provide a modified structured document. After the above-described modification, the method includes: (d) transforming the modified structured document into another document suitable for presentation; (e) displaying the other document suitable for presentation using the processing functionality to provide a displayed document; and (f) editing the displayed document.

The input structured document can be expressed in the extensible markup language (XML). The other document can be expressed in the hypertext markup language (HTML).

In one implementation, the operation of modifying is implemented using an upgrade module that provides a transformation function using extensible stylesheet language (XSL). *See Application, Page 4, Lines 5-16.*

The above-referenced determining of whether each of the particular features matches a set of expected features associated with the particular version of the processing functionality can include determining whether the input structured document contains each node expected by the particular version of the processing functionality. The above-referenced modifying of the particular features of the input structured document to produce the modified structured document can include: (cl) creating each node expected by the particular version of the processing functionality to provide created nodes; (c2) copying node content from the input structured document into corresponding created nodes in the modified structured document for those nodes in the input structured document that have counterpart nodes expected by the particular version of the processing functionality; and (c3) creating default node content in corresponding nodes in the modified structured document for those created nodes that do not have counterpart nodes in the input structured document. *See Application, Page 4, Lines 17-21.*

In one case, the above-referenced "expected features" are specified by a schema associated with the particular version of the processing functionality. In another case, the above-referenced "expected features" are specified by some information other than the schema associated with the particular version of the processing functionality (such as aspects pertaining to the visual presentation of the document that are not dictated by the schema). *See Application, Page 4, Line 22 to Page 5, Line 9.*

According to another exemplary implementation, a method is described for generating an upgrade module for upgrading documents for processing by processing functionality. The method includes: (a) determining whether a particular version of the processing functionality has been created that warrants generation of the upgrade module; and (b) generating the upgrade module if the creation of the particular version warrants the generation of the upgrade module. *See Application, Page 5, Lines10-15.*

Independent Claim 1 recites a method for upgrading documents for processing by processing functionality, comprising:

- inputting a structured document (e.g., reference number 202, FIG. 2; Page 9, Lines 24 to Page 10, Line 12) having particular editing controls associated therewith into a particular version of the processing functionality (e.g., Page 4, Lines 7-8);
- determining whether each of the particular editing controls matches a set of expected editing controls associated with the particular version of the processing functionality (e.g., page 4, lines 8-10; reference number 1208, FIG. 12; page 32, lines 10-15); and
- modifying the particular editing controls of the input structured document so that the particular editing controls match the set of expected editing controls to thereby

provide a modified structured document (e.g., page 4, lines 10-13; Page 13, Line 19 to Page 14, Line 9).

Independent Claim 20 recites a method for generating an upgrade module for upgrading documents for processing by processing functionality, comprising:

- determining whether a particular version of the processing functionality has been created that warrants generation of the upgrade module (e.g., Page 4, Lines 8-10; reference number 1208, FIG. 12; Page 32, Lines 10-15); and
- generating the upgrade module if the creation of the particular version warrants the generation of the upgrade module, wherein the upgrade module is configured to modify an input structured document having particular editing controls associated therewith to create an updated document which conforms to a set of expected editing controls associated with the particular version of the processing functionality (e.g., reference number 908, FIG. 9, Page 27, Lines 1-16; reference number 240, FIG. 2, Page 16, Lines 16-23).

Independent Claim 24 recites an apparatus for processing documents, comprising:

- an upgrade module configured to modify an input structured document having particular features associated therewith so that the input structured document conforms to a set of expected editing controls associated with a particular version of the apparatus, to thereby produce a modified structured document(e.g., Page 4, Lines 8-10; reference number 1208, FIG. 12; reference number 238, FIG. 2; Page 16, Lines 3-15; Page 32, Lines 10-15); and

- a transformation module configured to transform the modified structured document into another document suitable for presentation (e.g., reference number 218, FIG. 2; Page 10, Lines 13-25).

Independent Claim 25 recites an apparatus for generating an upgrade module for upgrading documents for processing by processing functionality, comprising:

- logic configured to determine whether a particular version of the processing functionality has been created that warrants generation of the upgrade module (e.g., reference number 908, FIG. 9, Page 27, Lines 1-16); and
- logic configured to generate the upgrade module if the creation of the particular version warrants the generation of the upgrade module, wherein the upgrade module is configured to modify an input structured document having particular editing controls associated therewith to create an updated document which conforms to a set of expected editing controls associated with the particular version of the processing functionality (e.g., reference number 240, FIG. 2, Page 16, Lines 16-23).

Independent Claim 26 recites a computer readable medium having stored thereon an information structure, comprising:

- an upgrade module information structure configured to modify an input structured document having particular editing controls associated therewith so that the input structured document conforms to a set of expected editing controls associated with a particular version of a processing apparatus, to thereby produce a modified structured document (e.g., Page 4, Lines 8-10; reference number 1208, FIG. 12; Page 32, Lines 10-15); and
- a transformation module information structure configured to transform the modified structured document into another document suitable for presentation (e.g., reference number 218, FIG. 2; Page 10, Lines 13-25).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-26 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Altova, Users Reference Manual Version 4.4 XML Spy Suite 4.4, Altova Ges. mbH & Altova, Inc., May 4, 2002 (hereinafter "Altova") in view of U.S. Patent Number 5,412,772 to Monson, L. (hereinafter "Monson").

ARGUMENT

Ground of Rejection Claims 1-26 satisfy the requirements of 35 U.S.C. § 103(a) and therefore are patentable over Altova in view of Monson.

A. Neither Altova nor Monson, alone or in combination, teach or suggest determining whether editing controls match a set of expected editing controls nor modification of the editing controls

Claim 1 in-part recites a method for upgrading documents for processing functionality, including

- inputting a structured document having particular editing controls associated therewith into a particular version of the processing functionality;
- determining whether each of the particular editing controls matches a set of expected editing controls associated with the particular version of the processing functionality; and
- modifying the particular editing controls of the input structured document so that the particular editing controls match the set of expected editing controls to thereby provide a modified structured document

The Examiner is correct that Altova fails to teach all the above features. However, the Examiner's citation of Monson as correcting these deficiencies is incorrect. The Monson reference simply teaches a system for permitting the viewing of an object in two different operating system environments. Monson, Abstract and Monson, Col. 3-4. Nowhere does the Monson reference teach or suggest the above method which is capable of modifying the particular editing controls . . . so that the particular editing controls match the set of expected editing controls to thereby provide a modified structured document.

The Monson reference also fails to "determine whether each of the particular editing controls matches a set of expected editing controls. . ." Monson does not teach or suggest this ability as all Monson is concerned (solely) with is taking an object (such as a textual group) and making that object viewable in a different operating system environment.

Thus, Monson does not have the capability of "modifying the particular editing controls". Instead, Monson is solely trying to provide the capability of presenting an object in a different environment. Monson only discloses that an arbitrary graphical view item may be placed in the "window." Monson, Col. 3, line 55 through Col. 4, line 4. This fails to teach the recited features as the Monson reference fails to teach or disclose inputting a structured document having particular editing controls, determining whether each of the particular editing controls matches a set of expected editing controls and modifying the particular editing controls of the input structured document. Altova in-view of Monson does not teach these features. Rather, one reading Altova/Monson would be instructed to utilize "user selected" graphical view items. The asserted combination does not teach or suggest the presently recited language. The Altova/Monson combination only teaches utilizing arbitrary graphical view items which have been predefined by a system administrator. Monson, Col. 4, lines 14-15. In other words, Monson modifies Altova by teaching that a person, such as a system administrator, may arbitrarily insert graphical view items. Monson, Col. 6, lines 21-35. Therefore, in the Altova/Monson combination, the end user is "stuck" with the arbitrary decision in comparison to the currently recited features. The Altova/Monson combination fails to disclose inputting, determining and modifying and therefore does not obviate the claim language. Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

Claims 3 and 4 are allowable based on their dependency from Claim 1. Additionally, Claims 3 and 4 are allowable for based on the individual claim's own recited features which are not disclosed by Altova/Monson. Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

Claim 7 is allowable based on its dependency from Claim 1, which is believed to be in a condition for allowance. Claim 7 is additionally allowable as neither the cited passage of Altova nor anywhere does the Altova/Monson references teach modifying, as recited in Claim 1, using an upgrade module that provides a transformation function using extensible stylesheet language (XSL). The references only teach XSL conversion rather than the recited methodology. Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

Claim 8 is allowable based on the same rationale as claim 1. Removal of the pending rejection is respectfully requested and allowance is solicited.

B. Neither Altova nor Monson, alone or in combination, teach or suggest generating an upgrade module

Independent **Claim 20** stands rejected over Altova in view of Monson. The Applicant disagrees. The Applicant respectfully notes that Claim 20 is directed (generally to) a method for generating an upgrade module and includes features which differ from Claim 1. Claim 20 recites the following in part:

- determining whether a particular version of the processing functionality has been created that warrants generation of the upgrade module; and

- generating the upgrade module if the creation of the particular version warrants the generation of the upgrade module, wherein the upgrade module is configured to modify an input structured document having particular editing controls associated therewith to create an updated document which conforms to a set of expected editing controls associated with the particular version of the processing functionality.

Altova in view of Monson fails to teach determining whether a particular version of the processing functionality . . . warrants generation of the upgrade module and generating the upgrade module . . . wherein the upgrade module is configured to modify an input structured document having particular editing controls associated therewith to create an updated document which conforms to a set of expected editing controls associated with the particular version of the processing functionality. Monson, asserted as correcting the deficiencies in Altova, does not teach these features.

Monson discloses a system in which human intervention is required to select view items for the resultant view. The Monson items are selected in an arbitrary manner. Monson, Col. 6, lines 21-40 "A user may select and position a view item by a method know in the art (e.g., "clicking" on an item using a mouse, and "dragging" the item to the desired position in the window). Monson Col. 3 line 64. Nowhere does the Monson reference teach or suggest the utilization of an upgrade module wherein the upgrade module is configured to modify an input structured document having particular editing controls associated therewith to create an updated document which conforms to a set of expected editing controls associated with the particular version of the processing functionality. The Monson disclosure is limited to providing a system for permitting view of textual objects in various operating system environments and does not disclose the generating of an update module as described. Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

C. Neither Altova nor Monson, alone or in combination, teach or suggest transformation of a modified structured document

Claim 2 is allowable based on its dependency from Claim 1. Moreover, nowhere does the Altova/Monson combination teach or suggest transforming the modified structured document . . . displaying the other document suitable for presentation using the processing functionality. Altova and Monson fail to transform a modified structured document. Instead the Altova/Monson combination only includes arbitrary graphical view items and does not meet the features of the present method. Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

D. Neither Altova nor Monson, alone or in combination, teach or suggest displaying the other document for presentation using the processing functionality to provide a displayed document nor expressing the other document in a markup language that uses tags pertaining to visual features

Claim 5 is allowable based on its dependency from Claim 2. The Altova/Monson combination fails to teach, among other features, displaying the other document for presentation using the processing functionality to provide a displayed document. Additionally, the cited portion of Altova asserted by the Examiner fails to teach the other document suitable for presentation is expressed in a markup language that uses tags pertaining to visual features associated with the presentation of the other document. The cited Altova passages fail to teach this feature. Rather, the cited portion of Altova requires an underlying XSL file. Altova, Page 70. The recited method overcomes this short-coming in the Altova reference.

Claim 5 in-part recites wherein the other document suitable for presentation is expressed in a markup language that uses tags. As Altova fails to teach or suggest the foregoing feature the combination does not teach each and every limitation as required. To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Ryoka*, 180 U.S.P.Q. 580 (C.C.P.A. 1974). *See also In re Wilson*, 165 U.S.P.Q. 494 (C.C.P.A. 1970). Removal of the pending rejection is requested as the Altova/Monson fails to teach wherein the other document suitable for presentation is expressed in a markup language that uses tags . . . Allowance of Claim 5 is earnestly solicited. **Claim 6** is allowable based on the same rationale as claim 5. Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

E. Neither Altova nor Monson, alone or in combination, teach or suggest determining whether each of the particular editing controls matches a set of expected editing controls associated with the particular version of the processing functionality

Claim 9 is allowable based on its dependency from Claim 1. The Applicant respectfully notes that the term "node set" as asserted in the pending Action does not appear in Claim 9. Claim 9 in-part recites "wherein the determining of whether each of the particular editing controls matches a set of expected editing controls associated with the particular version of the processing functionality . . ." Altova/Monson does not teach this feature as Monson, asserted as correcting the defects in Altova, simply teaches the utilization of arbitrary graphical view items. As a result of this teaching, Monson does not match a set of expected editing controls as only arbitrary selections are made. Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

Claim 10 is further allowable over it base Claim 9 as Altova in view of Monson fails to teach "copying editing control content from the input structured document into corresponding created editing controls in the modified structured document . . ." The citation of the Altova passage, in the rejection of Claim 10, is incorrect as the Office's rejection of Claim 1 specifically notes that it relies on Monson for the asserted teaching of modifying of editing controls. The entirety of the Office's argument under 103(a) is based on the need for Monson to correct the deficiency in Altova. Neither the cited passage of Altova nor Monson teach the "copying editing control content from the input structured document into corresponding created editing controls in the modified structured document . . ." As noted previously, Monson only teaches arbitrary insertion of graphical view items rather than copying. Based on this teaching, the Office's argument fails as the arbitrarily selected graphical items could not be copied as previously described. Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

Claim 13 is allowable based on its dependency from Claim 1. Claim 13 is additionally allowable as nowhere in the cited passage of Altova or anywhere in Altova or the other reference are expected editing controls specified by a schema. The cited passage of Altova does not disclose editing controls. The Altova passage, pages 186-188, simply teaches identifying data types in textual data and not editing controls as asserted. The Examiner's own rejection is based on the inclusion of Monson to correct the failure of Altova to teach "editing controls". Quite simply, Altova discloses identifying data types in textual data, i.e., identifying 8/1/2006 as a date. This disclosure does not directly or impliedly teach "editing controls" as asserted by the Examiner's argument. Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

Claim 14 in-part recites "wherein the expected editing controls are specified by some information other than a schema associated with the particular version of the processing functionality." Claim 14 is allowable as the Monson reference does not utilize "some information other than a schema" because, Monson is an arbitrary assignment, in which no information is utilized. Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

F. Neither Altova nor Monson, alone or in combination, teach or suggest determining whether the input structured document lacks editing controls or creating new editing controls . . . providing that the new editing controls are required in the particular version of the processing functionality

Claims 11 and 12 are allowable as nowhere does the Altova/Monson combination teach "determining whether the input structured document lacks editing controls. . ." (Claim 11) or "creating new editing controls . . . providing that the new editing controls are required in the particular version of the processing functionality. . ." (Claim 12). Altova/Monson does not teach this as this would be in contradiction to the arbitrary Monson system in which a system administrator selects graphical view items.

Claim 18 is allowable as depending from an independent Claim 1 which is in a condition for allowance. Claim 18 in part recites,

- creating new editing controls in the modified structured document providing that the new editing controls are lacking in the input structured document and providing that the new editing controls are required in the particular version of the processing functionality.

Nowhere does Altova/Monson teach or suggest creating new editing controls as recited. The Monson reference, asserted for this teaching, fails to teach the features of Claim 18. Monson discloses that graphical view items are selected by a user. Monson, Col. 6, lines 21-40. The selection is to occur in an arbitrary manner, such as by a user "dragging and dropping" the item. Thus, selection of the particular Monson graphical view items is conducted without regard for the necessity of the item. For example, through inadvertence, a user may forego a graphical view item which is required in the particular version of the processing functionality. As a result of this inadvertence, the resultant view, in Monson, may be deficient. In light of the foregoing, Altova/Monson fail to teach each and every feature and thus a *prima facie* case of obviousness does not exist. Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

Claims 21-22 are allowable based on their dependency from Claim 20. Additionally, each of the claims recite features which are not disclosed in the prior art. With regard to Claim 22, nowhere does Altova or Monson teach or disclose an upgrade module which "is configured to create new editing controls in the input structured document such that the updated document conforms to the set of expected editing controls associated with the particular version of the processing functionality." Monson does not teach this feature. Instead, Monson allows the user to arbitrarily select view items to be "dragged and dropped" in the view. Monson Col. 3 line 64. This fails to teach or suggest the recited feature as Monson does not teach conforming "to the set of expected editing controls associated with the particular version of the processing functionality." Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

G. Neither Altova nor Monson, alone or in combination, teach or suggest a document corresponding to a markup language document generated by an earlier version of the processing functionality compared to the particular version

Claims 15 and 16 are allowable based on their dependency from Claim 1. Claims 15 and 16 are additionally allowable as neither the cited portion of the Altova reference nor anywhere in Altova/Monson reference is a document corresponding "to a markup language document generated by an earlier version of the processing functionality compare to the particular version" taught or suggested. The cited passage simply identifies the version of the document and does not restrict the input structured document . . . by an earlier version of the processing functionality compared to the particular version. This is to say, Altova does not teach this feature as Altova merely permits version identification rather than the recited methodology. The same is true for Claim 16. "[I]t is necessary to ascertain whether the prior art teachings would appear to be sufficient to one of ordinary skill in the art to suggest making the claimed substitution or other modification." *In re Lalu*, 747 F.2d 703, 223 USPQ 1257, 1258 (Fed. Cir. 1984). Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

H. Neither Altova nor Monson, alone or in combination, teach or suggest an upgrade module

Claim 17 is allowable based on the same rationale as Claim 1. As the Altova/Monson combination fails to teach modifying as recited in Claim 1, the combination of Altova and Monson fails to teach or suggest using an upgrade module . . . which is developed without knowledge of any requirements of any input structured document (to perform modifying). Monson does not need this upgrade module as the selection is arbitrary. Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

I. Neither Altova nor Monson, alone or in combination, teach or suggest omitting editing controls that are not required

Claim 19 is allowable as Altova in view of Monson fails to teach the recited features. Monson, cited as correcting the deficiencies with respect to editing controls, fails to teach or suggest "omitting from the modified structured document existing editing controls in the input structured document that are not required in the particular version of the processing functionality." Monson does not implement a method having this feature as a system administrator is relied on to select the graphical view items to utilized in the resultant view. As such, the Monson system does not include the functionality of omitting . . . editing controls which are not required. The Monson system relies solely on the discretion of the person selecting the items and is therefore subject to the inclusion of items which are not required. Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

Claim 23 is additionally allowable as nowhere does Monson teach omitting editing controls . . . such that the updated document conforms to the set of expected editing controls associated with the particular version of the processing functionality. The Monson reference does not teach this feature. Instead, Monson teaches that a user, such as a system administrator, should arbitrarily select visual items for inclusion in the view. In Monson, no consideration is made for conforming the view items to expected items. As Monson does not correct this deficiency in Altova a prima facie case of obviousness does not exist and the rejection is untenable. Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

J. Neither Altova nor Monson, alone or in combination, teach or suggest an upgrade nor transformation module

Independent **Claim 24** stands rejected under 35 U.S.C. §103(a) over Altova in view of Monson. The Applicants disagree. Claim 24 generally recites an apparatus for processing documents and includes features which differ from those included in Claim 1. Claims 24 in part recites,

- an upgrade module configured to modify an input structured document having particular features associated therewith so that the input structured document conforms to a set of expected editing controls associated with a particular version of the apparatus, to thereby produce a modified structured document; and

- a transformation module configured to transform the modified structured document into another document suitable for presentation.

The Altova/Monson combination fails to teach these features. The asserted combination does not include an upgrade module nor do either of the references include a transformation module. Monson solely relies on the arbitrary decision of a user, such as a system administrator, to select view items. In light of this teaching, Monson does not correct the deficiency in Altova because there is no need for either an upgrade module or a transformation module as the view items are arbitrarily selected by the system administrator. Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

Claim 26 is an independent claim and recites features which differ from those recited in Claim 1 including an upgrade module and a transformation module. Neither of the foregoing features are recited in Claim 1 thus the reference to the rejection of Claim 1 is without merit. The Applicants disagree with the pending rejection. Claim 26 in part recites,

• an upgrade module information structure configured to modify an input structured document having particular editing controls associated therewith so that the input structured document conforms to a set of expected editing controls associated with a particular version of a processing apparatus, to thereby produce a modified structured document; and

• a transformation module information structure configured to transform the modified structured document into another document suitable for presentation.

As discussed previously, the asserted combination of Altova/Monson fails to teach or suggest computer readable medium having stored thereon an information structure including an upgrade module information structure. . . so that the input structured document conforms to a set of expected editing controls associated with a particular version of a processing apparatus. Instead, Monson, asserted as correcting the primary Altova reference, teaches a system in which users can select view items in order to have a "familiar" view. The Monson system makes no provision for conforming expected controls associated with a particular version of a processing apparatus. Monson teaches away from this capability as the Monson system is attempting to utilize a system administrator selected window for the various users instead of utilizing computer readable medium having stored thereon an information structure including an upgrade module information structure as recited. In the asserted Altova/Monson combination, a user is permitted to have an arbitrarily selected view which is familiar instead of implementing computer readable media in accordance with the present claim. Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

K. Neither Altova nor Monson, alone or in combination, teach or suggest logic configured to generate an upgrade module

Independent Claim 25 is allowable as neither the Altova reference nor the Monson reference alone or in combination establish a *prima facie* case of obviousness. Applicants disagree with the pending rejection. Claim 25 differs from Claim 20 and is separately patentable. Claim 25 is directed generally to an apparatus while Claim 20 is directed to a method. Claim 25 in part recites,

• logic configured to determine whether a particular version of the processing functionality has been created that warrants generation of the upgrade module; and

- logic configured to generate the upgrade module if the creation of the particular version warrants the generation of the upgrade module, wherein the upgrade module is configured to modify an input structured document having particular editing controls associated therewith to create an updated document which conforms to a set of expected editing controls associated with the particular version of the processing functionality.

The combination of Altova in view of Monson fails to teach these features. Rather, one reading Altova/Monson would be instructed to utilize arbitrary selection to insert view items instead of an apparatus including logic configured to determine whether a particular version of the processing functionality has been created that warrants generation of the upgrade module. Monson also fails to disclose an apparatus including logic configured to generate the upgrade module if the creation of the particular version warrants the generation of the upgrade module. The Monson reference is directed to permitting presentation of a "familiar" user defined view and is inconsistent with the features of Claim 20. Monson, Abstract and Monson Col. 6 lines, 10-16. At no time does Monson suggest conforms to a set of expected editing controls associated with the particular version of the processing functionality. The features of the instant claim are contrary to Monson as the claim recites conforming expected editing controls with the particular version of the processing functionality while Monson teaches that view items may be arbitrarily selected. Accordingly, the Applicant respectfully requests that the Board overturn this rejection.

CONCLUSION

The Applicant respectfully considers this application to be in condition for allowance and respectfully requests the Board to overturn the final rejection and that the Examiner pass this application to allowance.

Dated this 30th day of November, 2006.

Respectfully submitted,

WILLIAM J. BREEN, III
Attorney for Applicant
Registration No. 45,313

LEE & HAYES PLLC
421 W. Riverside Avenue, Suite 500
Spokane, WA 99201
Telephone: (509) 324-9256 (Ext. 249)
Facsimile: (509) 323-8979

APPENDIX: CLAIMS ON APPEAL

1. A method for upgrading documents for processing by processing functionality, comprising:

inputting a structured document having particular editing controls associated therewith into a particular version of the processing functionality;

determining whether each of the particular editing controls matches a set of expected editing controls associated with the particular version of the processing functionality; and

modifying the particular editing controls of the input structured document so that the particular editing controls match the set of expected editing controls to thereby provide a modified structured document.

2. A method according to claim 1, further comprising:

transforming the modified structured document into another document suitable for presentation;

displaying the other document suitable for presentation using the processing functionality to provide a displayed document; and

editing the displayed document.

3. The method according to claim 1, wherein the input structured document is expressed in a markup language that uses tags pertaining to subject matter fields in the input structured document.

4. The method according to claim 3, wherein the input structured document is expressed in the extensible markup language (XML).

5. The method according to claim 2, wherein the other document suitable for presentation is expressed in a markup language that uses tags pertaining to visual features associated with the presentation of the other document.

6. The method according to claim 5, wherein the other document suitable for presentation is expressed in the hypertext markup language (HTML).

7. The method according to claim 1, wherein the modifying uses an upgrade module that provides a transformation function using extensible stylesheet language (XSL).

8. The method according to claim 2, wherein the other document suitable for presentation comprises an electronic form having at least one user data entry field therein.

9. The method according to claim 1, wherein the determining of whether each of the particular editing controls matches a set of expected editing controls associated with the particular version of the processing functionality comprises:

determining whether the input structured document contains each editing controls expected by the particular version of the processing functionality.

10. The method according to claim 9, wherein the modifying of the particular editing controls of the input structured document to produce the modified structured document comprises:

creating each editing controls expected by the particular version of the processing functionality to provide created editing controls;

copying editing controls content from the input structured document into corresponding created editing controls in the modified structured document for those editing controls in the input structured document that have counterpart editing controls expected by the particular version of the processing functionality; and

creating default editing controls content in corresponding editing controls in the modified structured document for those created editing controls that do not have counterpart editing controls in the input structured document.

11. The method according to claim 1, wherein the determining of whether each of the particular editing controls matches a set of expected editing controls associated with the particular version of the processing functionality comprises:

determining whether the input structured document lacks editing controls that were previously classified as optional but are no longer classified as optional in the particular version of the processing functionality.

12. The method according to claim 11, wherein the modifying of the particular editing controls of the input structured document to produce the modified structured document comprises:

creating new editing controls in the modified structured document providing that the new editing controls are lacking in the input structured document and providing that the new editing controls are required in the particular version of the processing functionality although considered optional by its schema.

13. The method according to claim 1, wherein the expected editing controls are specified by a schema associated with the particular version of the processing functionality.

14. The method according to claim 1, wherein the expected editing controls are specified by some information other than a schema associated with the particular version of the processing functionality.

15. The method according to claim 1, wherein the input structured document corresponds to a markup language document generated by an earlier version of the processing functionality compared to the particular version.

16. The method according to claim 1, wherein the input structured document corresponds to a markup language document generated by a later version of the processing functionality compared to the particular version.

17. The method according to claim 1, wherein the modifying is performed using an upgrade module, and wherein the upgrade module is developed without knowledge of any requirements of any input structured document.

18. The method according to claim1, wherein modifying of the particular editing controls of the input structured document to produce the modified structured document comprises:

creating new editing controls in the modified structured document providing that the new editing controls are lacking in the input structured document and providing that the new editing controls are required in the particular version of the processing functionality.

19. The method according to claim 1, wherein modifying of the particular editing controls of the input structured document to produce the modified structured document comprises:

omitting from the modified structured document existing editing controls in the input structured document that are not required in the particular version of the processing functionality.

20. A method for generating an upgrade module for upgrading documents for processing by processing functionality, comprising:

determining whether a particular version of the processing functionality has been created that warrants generation of the upgrade module; and

generating the upgrade module if the creation of the particular version warrants the generation of the upgrade module, wherein the upgrade module is configured to modify an input structured document having particular editing controls associated therewith to create an updated document which conforms to a set of expected editing controls associated with the particular version of the processing functionality.

21. The method of claim 20, wherein the upgrade module is formed using the extensible stylesheet language (XSL).

22. The method according to claim 20, wherein the updated module is configured to create new editing controls in the input structured document such that the updated document conforms to the set of expected editing controls associated with the particular version of the processing functionality.

23. The method according to claim 20, wherein the updated module is configured to omit editing controls in the input structured document from updated document such that the updated document conforms to the set of expected editing controls associated with the particular version of the processing functionality.

24. An apparatus for processing documents, comprising:

an upgrade module configured to modify an input structured document having particular features associated therewith so that the input structured document conforms to a set of expected editing controls associated with a particular version of the apparatus, to thereby produce a modified structured document; and a transformation module configured to transform the modified structured document into another document suitable for presentation.

25. An apparatus for generating an upgrade module for upgrading documents for processing by processing functionality, comprising:

logic configured to determine whether a particular version of the processing functionality has been created that warrants generation of the upgrade module; and

logic configured to generate the upgrade module if the creation of the particular version warrants the generation of the upgrade module, wherein the upgrade module is configured to modify an input structured document having particular editing controls associated therewith to create an updated document which conforms to a set of expected editing controls associated with the particular version of the processing functionality.

26. A computer readable medium having stored thereon an information structure, comprising:

an upgrade module information structure configured to modify an input structured document having particular editing controls associated therewith so that the input structured document conforms to a set of expected editing controls associated with a particular version of a processing apparatus, to thereby produce a modified structured document; and a transformation module information structure configured to transform the modified structured document into another document suitable for presentation.

APPENDIX: EVIDENCE

None.

APPENDIX: RELATED PROCEEDINGS

None.